

Uso de Métricas para la Gestión de Riesgos

Cecilia Belletti cecibell@adinet.com.uy / 3967@universidad.ort.edu.uy

Luis Jaunarena luisj@adinet.com.uy / 101915@universidad.ort.edu.uy

Montevideo, Uruguay

Resumen

A partir de la aplicación de algunos de los conceptos de los modelos teóricos de calidad de software y las metodologías de gestión de riesgos y métricas, se plantea una metodología que combina lo anterior, y se presenta un marco para utilizarse como herramienta para la evaluación y seguimiento de riesgos.

Los objetivos de esta propuesta, consisten en proveer a los gerentes de proyectos de una herramienta para el seguimiento efectivo y eficaz de los riesgos más importantes a lo largo del proyecto, y un método sistemático para realizar dicho seguimiento, así como también vincular los procesos de calidad y gestión de riesgos de los proyectos de software.

Palabras clave

Calidad, Métricas, Riesgos, SQA.

1. Introducción

Los avances en la ingeniería de software, han resultado en una variada gama de procesos, modelos y metodologías para ser tomados como referencia al definir el proceso de software para un proyecto. Dichos modelos de procesos deben poder alterarse y adaptarse para satisfacer las necesidades y propósitos de los proyectos específicos, y deben ofrecer facilidad de planificación, gestión, ejecución y control [Pressman, 1998]. A su vez, los procesos de las distintas áreas que intervienen y conforman un proyecto, deben actuar interactiva y coordinadamente, siendo en muchos casos las salidas de algunos procesos de un área, la entrada de procesos de otra área. En este artículo se mencionan y analizan algunos modelos teóricos de calidad de software y metodologías de gestión de riesgos y métricas, y se plantea un modelo de evaluación de riesgos utilizando métricas, que combina las metodologías básicas de gestión de riesgos y de ciclo de métricas.

La estructura de este artículo comienza con una presentación del concepto de calidad visto desde la óptica de distintos autores (Sección 2) y una introducción a las métricas, haciendo énfasis en porqué son necesarias, cuáles son sus usos, qué es importante medir, y se presentan algunos de los modelos de calidad más conocidos (Sección 3). La Sección 4 introduce al lector a la Gestión de Riesgos, y en la siguiente sección, se mencionan las metodologías utilizadas para las métricas y para la gestión de riesgos, y se realiza un breve análisis de dichas metodologías. En las secciones siguientes, se propone un modelo de evaluación de riesgos y la metodología que combina la utilización de métricas y la gestión de riesgos (Secciones 5 y 6). En la sección 8 del artículo, se presenta un caso de estudio donde se aplicó dicho marco.

2. Calidad

J. M. Juran define el Aseguramiento de la Calidad como sigue [Schulmeyer, 1999]:

El Aseguramiento de la Calidad es la actividad que provee a todos los involucrados, la evidencia necesaria para establecer la confianza de que la función de la calidad está siendo realizada adecuadamente.

A su vez, la IEEE provee la siguiente definición de esta actividad [Schulmeyer, 1999]:

El Aseguramiento de la Calidad es un patrón planificado y sistematizado de todas las acciones necesarias para proveer la confianza adecuada de que un ítem o producto conforma los requerimientos técnicos establecidos.

La definición anterior hace énfasis en el producto. Definiciones más modernas incluyen actividades que aseguren tanto la calidad del producto como la del *proceso*. Garvin describe la calidad desde cinco perspectivas diferentes: la vista trascendental, que ve la calidad como algo que puede ser reconocido pero no definido; la vista del usuario, que ve la calidad como el cumplimiento de sus propósitos; la vista del productor, que ve la calidad como la conformidad con las especificaciones; la vista del producto, que ve la calidad unida a las características inherentes del producto; y la vista basada en el valor, que ve la calidad dependiente de cuánto está dispuesto el cliente a pagar por ella [Hyatt, 1996].

En un proyecto de software, para que se cumplan los objetivos de calidad, las actividades de aseguramiento de la calidad no deben ser aisladas, sino que deben estar relacionadas e inmersas en las distintas actividades del proyecto, siendo muy importante la interacción entre el gerente del proyecto y los encargados del área de SQA.

3. Métricas

Las mediciones se han convertido en un elemento esencial de las prácticas de ingeniería de software. Son utilizadas para verificar si los requerimientos son consistentes y completos, o si el código producido está listo para ser probado. A nivel gerencial, se miden atributos de proceso y de producto para controlar agenda y presupuesto y los clientes miden aspectos del producto final para determinar si cumple con sus expectativas [Fenton, 1997].

Una *medición* es el proceso por el cual números o símbolos son asignados a atributos de entidades del mundo real, de tal forma que los describen de acuerdo a reglas claramente definidas. Una entidad es un objeto o un evento del mundo real como puede ser una persona, o una fase de un proyecto. Un atributo es una característica o propiedad de una entidad. Las mediciones son correspondencias del mundo empírico a un sistema formal, y las medidas son los números o símbolos asignados usando esta correspondencia. Existen dos tipos de cuantificaciones, las mediciones y los cálculos. Mientras las mediciones son una cuantificación directa, los cálculos son indirectos, dado que son resultado de la combinación de mediciones. Las mediciones indirectas son útiles para hacer visible la interacción entre mediciones directas [Fenton, 1997].

Una *métrica* es una medida cuantitativa del grado en que un sistema, componente o proceso, posee un atributo dado. Un *indicador* es una variable que puede ser configurada con un estado señalado basándose en los resultados de un proceso o de la ocurrencia de una condición especificada. Un indicador generalmente compara una métrica con una línea base o un resultado esperado. Esto permite a la hora de tomar una decisión, una comparación que puede proveer una perspectiva acerca de algún aspecto del proyecto [Schulmeyer, 1999].

3.1. Por qué son necesarias las Métricas

En la mayoría de los proyectos de software, no se definen metas mensurables para los productos. Como resultado, cuando el proyecto llega a su fin, no se puede afirmar si se llegaron o no a cumplir los objetivos planteados. Otro problema surge al cuantificar los componentes de costo de un proyecto, no pudiéndose controlar el presupuesto, si no se miden los componentes relativos del costo. El no cuantificar o predecir la calidad de los productos impide mostrar al cliente en qué grado se alcanzaron sus requerimientos de calidad. Por otro lado, las decisiones sobre la adopción de nuevos procesos o tecnologías, no son siempre tomadas luego de realizar estudios cuidadosos y controlados para determinar si dicho nuevo proceso o tecnología es eficiente y eficaz [Fenton, 1997].

Las mediciones son importantes para comprender lo que está sucediendo durante el proceso de desarrollo, permitiendo conocer la situación actual, haciendo más visibles aspectos del proceso y del producto, otorgando mayor comprensión de las relaciones entre las actividades y las entidades que éstas afectan. A su vez, permiten controlar lo que está pasando en el proyecto, dado que permiten predecir lo que es probable que suceda y realizar cambios a los procesos y productos de modo que se cumplan las metas trazadas. Por otro lado, promueven la mejora de los procesos y productos [Fenton, 1997].

3.2. Usos de las Métricas

Las métricas de software abarcan varias actividades del proceso de desarrollo. A partir de la necesidad de predecir costos durante etapas tempranas de los proyectos, se han propuesto con el tiempo varios modelos de estimación de costo y esfuerzo, como por ejemplo, los modelos COCOMO y COCOMO 2.0 de Boehm [Boehm, 1995], el modelo SLIM de Putnam y los puntos funcionales de Albrecht [Pfleeger, 1998]. También las necesidades gerenciales han resultado en varios intentos de definir medidas y modelos de productividad. Es importante tener en cuenta, que la calidad de las mediciones depende fuertemente de una recolección cuidadosa de los datos, sobretodo cuando los mismos deben ser recolectados y compilados de diversos proyectos. Por otro lado, las métricas no tienen sentido si son evaluadas en forma aislada. Esto ha llevado al desarrollo de modelos de calidad, como los propuestos por Boehm y McCall que se verán en más detalle en la sección 3.4. Como ya se mencionó antes, las mediciones son necesarias al momento de evaluar nuevos métodos y herramientas. [Fenton, 1997].

En conclusión, las métricas son útiles para establecer objetivos de mejora cuantitativos, para la planificación de actividades, gestión de recursos humanos y de presupuesto. Es decir, son útiles para la gestión de proyectos, al colaborar con la toma de decisiones [Möller, 1993].

3.3. Qué medir

Una medición es útil sólo si ayuda a entender el proceso subyacente o uno de sus productos resultantes. El reconocimiento de la mejora de procesos y productos puede ocurrir sólo cuando se han definido claramente los objetivos para los procesos y productos. Es decir, no se puede decir si se está yendo en la dirección correcta, hasta que se sepa hacia dónde se quiere ir [Fenton, 1997].

Por lo tanto, identificar las entidades y atributos que se desea medir, es el primer paso de cualquier actividad de medición. Los mismos se pueden clasificar en tres categorías: *productos*, *procesos* y *recursos*. Además, se distingue entre atributos *internos* y *externos* de estas entidades. Los internos,

se pueden medir en términos del producto, proceso o recurso, independientemente de su ambiente; mientras que los externos, son los que se pueden medir con respecto a cómo el producto, proceso o recurso se relaciona con su entorno [Fenton, 1997].

Como *métricas de proceso* directas, se pueden mencionar la duración del proceso o de sus actividades, esfuerzo, cantidad de incidentes de algún tipo especificado. Muchas de estas medidas se pueden usar en combinación con otras para ganar mayor comprensión sobre lo que está ocurriendo en el proyecto, como ser los costos relativos a la calidad y la productividad [Fenton, 1997]. Las *métricas de producto* abarcan los atributos de calidad definidos para los mismos, y los problemas y defectos que estos presentan. Además, es importante medir la satisfacción de los clientes y usuarios del sistema.

3.4. Modelos de Calidad de Software

Dado que la calidad es un concepto compuesto, usualmente se captura en un modelo que muestra las relaciones entre las características. Tanto Boehm como McCall desarrollaron modelos de calidad en 1978 y 1977 respectivamente. En este tipo de modelos, se centraron en el producto final, identificando atributos clave de calidad desde la perspectiva del usuario. Estos atributos en general son atributos externos de alto nivel, como “usabilidad” o “mantenibilidad”, los cuales no se pueden medir directamente, así que se descomponen en atributos de más bajo nivel, que suelen ser atributos internos. Un nivel más profundo de descomposición, se da asociando los atributos antes mencionados con atributos mensurables, llamados métricas de calidad. Para monitorear la calidad, se puede optar por adaptar alguno de estos modelos, ya sea utilizando un subconjunto de los atributos que se proponen, o bien, siguiendo la idea general, pero definiendo específicamente la descomposición, de acuerdo a las necesidades del proyecto.¹ En 1992, se propuso el modelo de calidad estándar ISO 9126 (una derivación del modelo de McCall) marcando un hito importante ya que ofrece las ventajas de un modelo universal. El mismo descompone la calidad en seis factores: funcionalidad, confiabilidad, eficiencia, usabilidad, mantenibilidad y portabilidad [Fenton, 1997]. En la Figura 1, se muestra una tabla comparativa de los tres modelos de calidad presentados.

Objetivo	McCall (1977)	Boehm (1978)	ISO 9126 (1992)
Correctitud	X	X	Mantenibilidad
Confiabilidad	X	X	X
Integridad	X	X	
Usabilidad	X	X	X
Eficiencia	X	X	X
Facilidad de mantenimiento	X	X	X
Facilidad de prueba	X		Mantenibilidad
Interoperabilidad	X		
Flexibilidad	X	X	
Reusabilidad	X	X	
Portabilidad	X	X	X
Claridad		X	
Facilidad de modificación		X	Mantenibilidad
Documentación		X	
Facilidad de comprensión		X	
Validez		X	Mantenibilidad
Funcionalidad			X
Generalidad		X	
Economía		X	

Figura 1: Modelos de Calidad de Software [Hyatt, 1996]

¹ Ver en la sección 5, una metodología utilizada de la IEEE, que propone un marco de métricas de calidad de software.

4. Gestión de Riesgos

La Gestión de Riesgos es una disciplina cuyos objetivos son identificar, tratar y eliminar los elementos de riesgo antes de que se transformen en amenazas al éxito del proyecto [Boehm, 1989]. Provee un ambiente disciplinado para la toma de decisiones proactiva para revisar sistemáticamente qué puede ir mal (riesgos), determinar cuáles son los riesgos que es importante tratar, e implementar estrategias para tratar dichos riesgos [SEI, 2002].

El PMBOK [PMI, 1997] describe cuatro procesos dentro de la gestión de riesgos: identificación de riesgos, donde se determinan los riesgos que pueden llegar a afectar el proyecto y se documentan las características de cada uno; cuantificación de los riesgos, que consiste en la evaluación de los riesgos y sus interacciones para valorar el posible rango de posibles resultados; desarrollo de respuesta a los riesgos, para definir mejoras para las oportunidades y respuestas a las amenazas; y control de los cambios de los riesgos a lo largo del curso del proyecto. Estos procesos interactúan entre sí, y con otros procesos del proyecto, y cada uno puede requerir el esfuerzo de uno o más recursos.

4.1. Por qué es necesaria la Gestión de Riesgos

Tom Gilb, en su artículo *Principles of Software Engineering Management* (1988) [Boehm, 1989] presenta algunos principios que son útiles para comprender porqué es importante la gestión de riesgos en los proyectos:

Principio del Riesgo: Si los riesgos no se atacan activamente, ellos te atacarán activamente.

Principio de compartir el Riesgo: El profesional real es el que conoce los riesgos, su grado, sus causas, y las acciones necesarias para medirlos, y comparte su conocimiento con sus colegas y clientes.

Principio de la solicitud: Si no se busca información sobre los riesgos, se está en busca de problemas.

Principio de la exposición del Riesgo: El grado de un riesgo y sus causas, nunca debe ser ocultado a quienes deben tomar decisiones.

5. Metodologías utilizadas

Presentaremos a continuación, algunas de las metodologías ampliamente utilizadas para las métricas de calidad de software y para la gestión de riesgos.

5.1. Métricas de Calidad de Software

La metodología de métricas de calidad de software propuesta por IEEE *Standard for a Software Quality Metrics Methodology* (1993) propone un enfoque sistemático para establecer requerimientos de calidad e identificar, implementar, analizar y validar las métricas para un sistema de software. Abarca todo el ciclo de vida del proyecto y se compone de cinco pasos [Schulmeyer, 1999]:

Establecer los requerimientos de calidad de software: La metodología comienza identificando los requerimientos que pueden ser aplicables al sistema de software. Para crear esta lista se debe utilizar la experiencia organizacional y los estándares y regulaciones requeridos. También es importante listar otros requerimientos del sistema que puedan afectar los requerimientos de calidad. Se deben considerar aspectos como restricciones de costo y agenda, garantías e intereses organizacionales. Estos requerimientos se deben ordenar de acuerdo a su importancia, en función con las características del sistema y los puntos de vista de los agentes involucrados. Para cada factor, se debe asignar una métrica directa para representarlo, y un valor para servir como requerimiento cuantitativo de dicho valor. Por ejemplo, el requerimiento es “alta eficiencia”, con la métrica directa “recursos utilizados” y un valor de 90% que represente dicho factor. Sin este valor, no se dispone de un estándar que indique si el sistema alcanza o no este requerimiento. Un esquema de este modelo se presenta en la Figura 2.

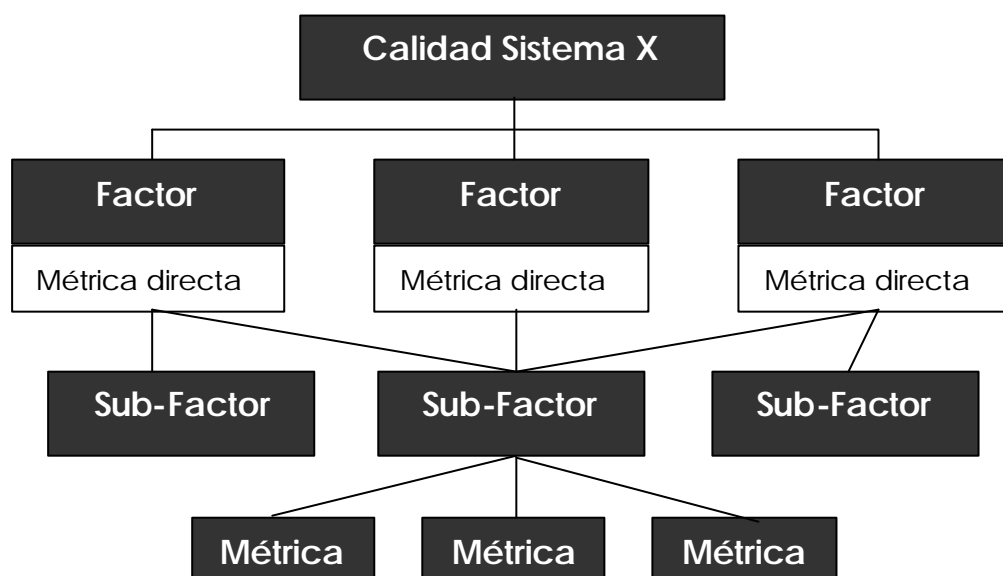


Figura 2: IEEE Standard for a Software Quality Metrics Methodology (1993)

Identificar las métricas de calidad de software: De acuerdo con los requerimientos de calidad se crea una tabla jerárquica con estructura de árbol, como se ve en la figura. En este punto, sólo el nivel FACTOR debería estar completo. Entonces, se descompone cada factor en sub-factores para cuantos niveles sean necesarios hasta que el sub-factor esté completo. En este momento, los sub-factores se descomponen en métricas usadas para medir los Marco de productos y procesos del sistema durante el ciclo de vida del proyecto. Luego de aplicado este marco, se debe realizar un análisis de costo-beneficio, para ajustar el conjunto de métricas de acuerdo al mismo.

Implementar las métricas de calidad de software: Para cada métrica del conjunto, se debe determinar los datos que deben ser recolectados y las asunciones hechas sobre los datos. A su vez se debe describir cuándo y cómo se utilizarán las herramientas, identificando las entidades que participan en la recolección. Los datos se deben recolectar en los momentos apropiados del ciclo de vida y se deben computar los valores de las métricas de acuerdo a los datos recolectados.

Analizar los resultados: Los resultados deberían ser interpretados y registrados tomando en cuenta el contexto del proyecto así como un producto o proceso particular.

Validar las métricas: El propósito es identificar ya sea métricas de producto como de proceso que puedan predecir valores de factores especificados, los cuales son representaciones cuantitativas de los requerimientos de calidad. Para que las métricas sean útiles, deben indicar con precisión si los requerimientos de calidad han sido alcanzados o si serán alcanzados en el futuro. Para ser considerada válida, una métrica debe demostrar un alto grado de asociación con los atributos de calidad que representa. Una métrica puede ser válida de acuerdo a cierto criterio e inválido con respecto a otro. Es necesario revalidar una métrica predictiva cada vez que es usada.

En el diagrama de la Figura 3 se muestra un resumen en líneas generales, del ciclo de vida de una métrica:

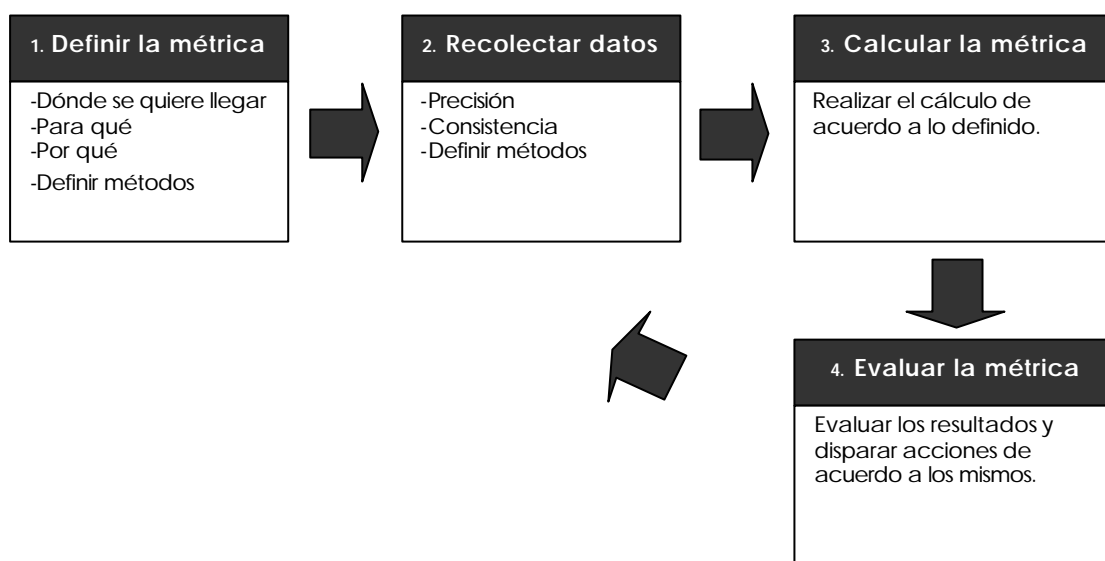


Figura 3: Ciclo de vida de una métrica

5.2. Gestión de Riesgos

Dale Karolak [Karolak, 1996] presenta una técnica para desarrollar métricas de riesgos de software, presentando las siguientes actividades:

Identificación de Riesgos: Karolak propone que a través de un cuestionario se identifiquen una serie de riesgos potenciales, organizándolos en distintas categorías, las cuales abarcan las distintas variables del proyecto como ser agenda, costos, alcance, organización, calidad, etc. Luego los riesgos identificados se ponderan y se analizan.

Análisis y Proyección de Riesgos: A partir la de la lista de riesgos identificados, éstos se analizan de acuerdo a su clasificación (técnicos, de RRHH, etc.), su impacto (agenda, esfuerzo, etc.), su ocurrencia en el tiempo (cercano, mediano o lejano plazo) y su probabilidad de ocurrencia. Para priorizar los riesgos se calcula su magnitud (producto de la severidad del impacto por la probabilidad de que ocurra). De este modo se obtiene una lista de los riesgos ordenados por su magnitud.

Estrategia y Planificación de Riesgos: Para los riesgos más importantes, se desarrolla una estrategia y planificación, que incluye un análisis detallado del mismo, actividades preventivas y correctivas y detalle del seguimiento. Se planifican las actividades de seguimiento, que incluyen nuevas evaluaciones de la magnitud y de los resultados de las actividades preventivas o correctivas llevadas a cabo.

Seguimiento: El seguimiento se debe realizar de acuerdo a la planificación realizada.

Algunos años antes, Barry Boehm desarrolló un marco para la gestión de riesgos de software presentando un conjunto de métodos acerca de la valoración y control de los riesgos [Boehm, 1989] [Oman, 1997]. El proceso de gestión de riesgos se puede resumir como la identificación de riesgos, priorización y monitoreo.

5.3. Análisis

Existe un número de dificultades en la aplicación directa de los modelos de calidad presentados arriba. Varios de los atributos sugeridos, en situaciones particulares, están incluidos en los requerimientos funcionales, de desempeño y de interface del sistema. Por otro lado, los modelos son estáticos, es decir, no describen cómo proyectar las métricas de los valores actuales a los valores de los hitos siguientes. Las proyecciones son necesarias para determinar el riesgo de los atributos que alcancen los criterios de éxitos establecidos. También es importante poder relacionar las métricas de software con el avance del proyecto y con valores esperados al momento de liberar el software. Finalmente, los modelos no proveen una guía para el uso de estas métricas y atributos en la gestión de riesgos [Hyatt, 1996].

Los modelos de gestión de riesgos de Boehm y Karolak son bastante similares en su enfoque general. Ambos consideran los riesgos en el proceso de software dentro un proceso planificado y plantean procesos de gestión que contienen un ciclo de pasos que se repiten a lo largo del proyecto. Estos pasos se centran en la identificación y análisis de los riesgos, haciendo énfasis en la documentación, la cual sirve como medio de comunicación a los involucrados. Por su parte el modelo de Karolak contiene una taxonomía de riesgos y métricas de riesgos.

Estos marcos de gestión de riesgos han demostrado ser muy útiles. Sin embargo presentan un mundo más predecible del que se puede encontrar en la realidad. En general, las crisis son dadas por lo inesperado, y es imposible proveer una lista o detalle de todos los riesgos posibles. Muchos riesgos, no son definibles hasta que la crisis llega. Además, más allá de una buena planificación de riesgos, los gerentes deben saber desenvolverse en estas situaciones. Es decir, además de ser buenos administrando el riesgo, deben ser capaces de manejar los problemas en forma reactiva.

6. Modelo de Evaluación de Riesgos

En las secciones anteriores, se mostraron y se analizaron distintos aspectos sobre la gestión de riesgos y la utilización de métricas. En esta sección, se propone un modelo de evaluación de riesgos, el cual toma elementos de las metodologías y marcos antes mencionados y cuyo esquema se muestra en la Figura 4.

El objetivo de este modelo, es ofrecer a los gerentes de proyecto, una herramienta para el seguimiento de los riesgos más importantes basándose en una evaluación objetiva y efectiva de los mismos. Los elementos que lo componen son los factores de riesgo, los elementos de riesgo propiamente dichos, y las métricas que se utilizan para evaluarlos.

Una vez identificados los factores de riesgo más importantes, y los posibles riesgos que generan, se seleccionan o definen métricas que sean útiles para conocer la evolución de cada elemento de riesgo identificado. Puede darse que estas métricas ya estén definidas dentro del modelo de calidad usado en el proyecto. Es importante establecer los rangos de valores esperados, para poder realizar las

comparaciones y así visualizar si el riesgo está evolucionando favorablemente, si están dando resultado las acciones preventivas, etc. En muchos casos, estos valores serán los establecidos dentro del modelo de calidad, pero esto puede llegar a variar.

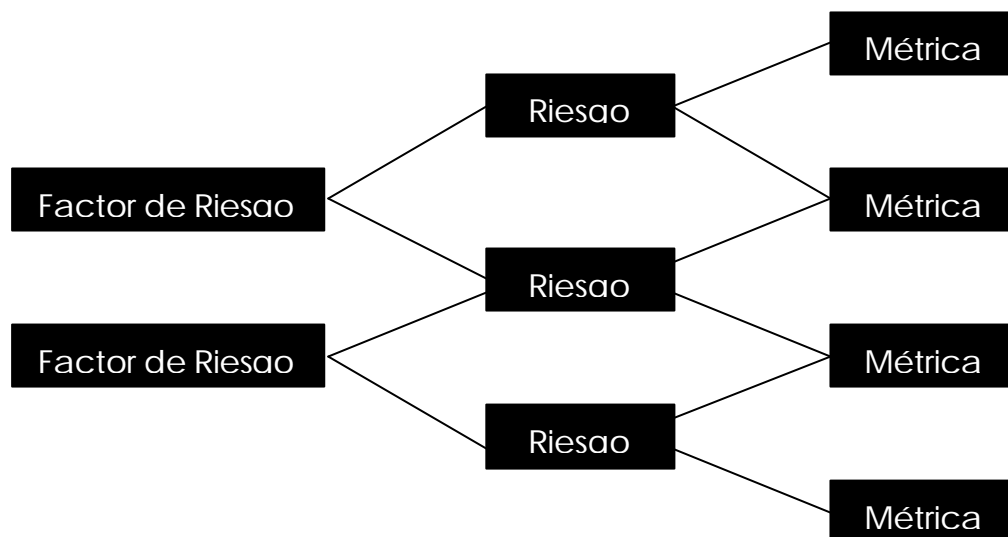


Figura 4: Modelo de Evaluación de Riesgos

De la misma forma que ocurre con los atributos de calidad, no siempre es sencillo encontrar, o recolectar datos para una métrica que responda a las necesidades. En estos casos se deberá analizar la relación costo – beneficio, y si el último es pobre o el primero muy alto, es conveniente considerar realizar el seguimiento del riesgo utilizando otros métodos.

Si bien este marco se complementa con modelos como el de Karolak, dado que se puede establecer un mapeo entre los resultados obtenidos y el impacto, no está orientado a una metodología en particular para el seguimiento y evaluación de riesgos. Sin embargo, debe existir esta relación (mapeo), dado que sin la misma, pierde sentido el cálculo de la métrica, porque no se tiene contra qué comparar [Fenton, 1997]. Es decir, se deben poder plantear objetivos específicos que cumplir en las evaluaciones, como por ejemplo: el riesgo X se mitigará si se logra que la productividad aumente en un 25%.

7. Metodología

Para poder utilizar el modelo presentado en la sección anterior, es imprescindible que exista una sincronización entre los procesos de gestión de riesgos y de métricas. Esto implica un trabajo conjunto del gerente del proyecto con el líder de SQA, al momento de realizar la planificación de métricas y la coordinación de tareas de recolección de datos, seguimiento, etc.

En la Figura 5, a continuación, se presenta una metodología que combina las metodologías básicas de gestión de riesgos, y ciclo de vida de las métricas.

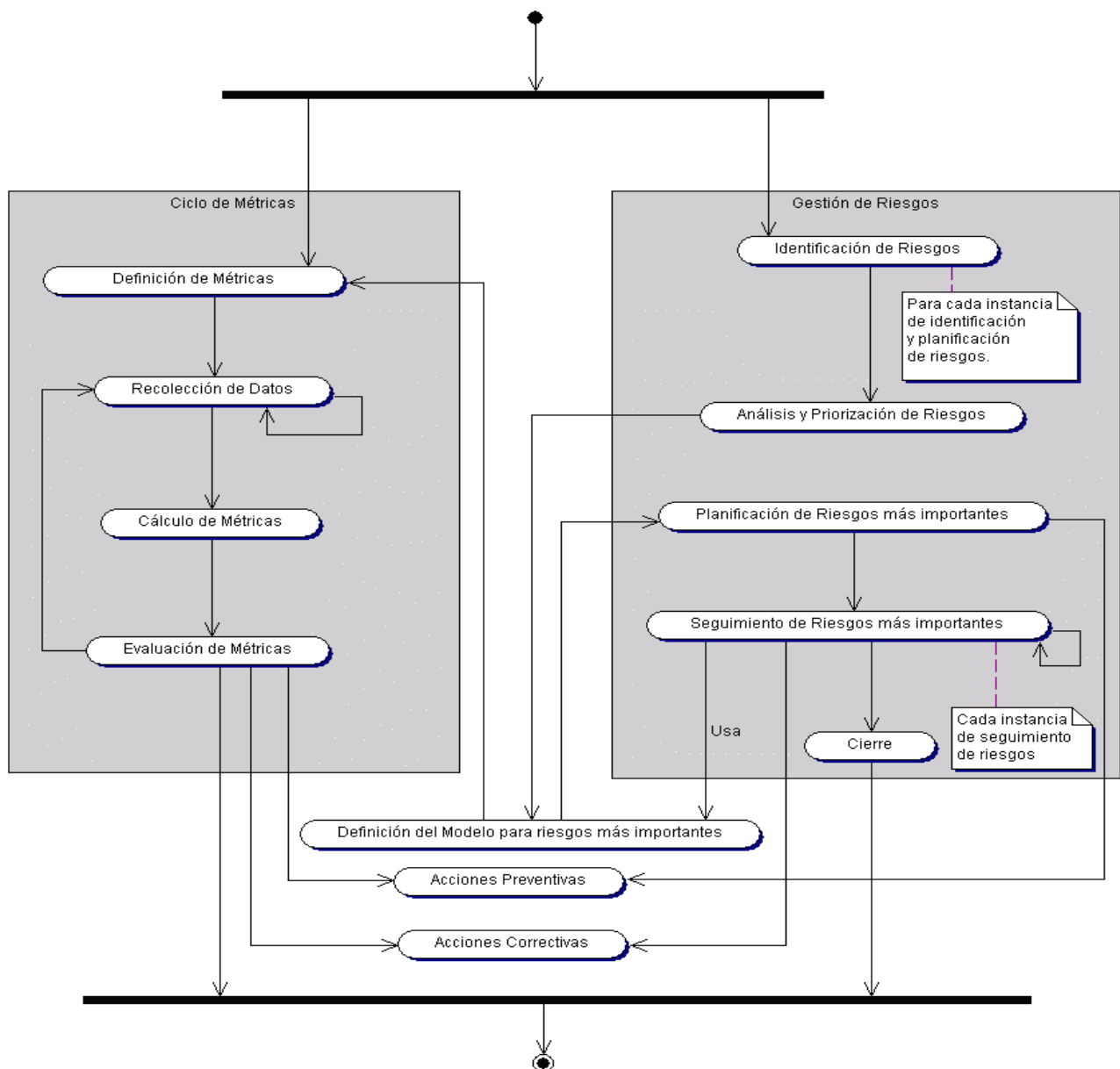


Figura 5: Metodología de Evaluación de Riesgos

Howard Rubin sostiene que *"Cada medición, debe estar vinculada con una necesidad"* [Oman, 1997]. Por tanto, la definición de métricas, no es un proceso que se realiza en un único momento del proyecto. A su vez, no debe ser una tarea exclusiva del área de SQA. Una posibilidad consiste en, al comienzo, de acuerdo a los requerimientos que se posean hasta el momento, definir un conjunto de atributos y por tanto, conjunto de métricas básico, y a partir del mismo, determinar los datos que se van a recolectar. Es importante, procurar recolectar un conjunto predeterminado de datos y realizar la menor cantidad de cambios posibles en las solicitudes, para evitar los problemas que se dan en la recolección de métricas, como por ejemplo, la resistencia a informar por parte de los integrantes de los equipos de trabajo, o falta de coordinación. Una vez definido este conjunto básico de métricas y de datos, a medida que surjan las necesidades se va completando el modelo de calidad y definiendo nuevas métricas.

Con respecto a la gestión de riesgos, durante el proceso de identificación, surgen distintos factores de riesgo que pueden llevar a muchos posibles riesgos que no se sabe en ese momento, cuáles pueden llegar a ser. Por otro lado, como se planteó antes, no se puede abarcar todos los riesgos posibles que existen en el universo. De este modo, es importante definir el alcance que se dará a la identificación y planificación de riesgos, y moverse dentro de ese alcance. Y además, es conveniente, no perder de vista desde el comienzo del proyecto, un conjunto de riesgos que por su importancia o las características del proyecto, sea importante gestionar. Boehm propone, por ejemplo, un “*Top 10*” de los elementos de riesgo [Oman, 1997].

Una vez identificados los riesgos más importantes, se construye el modelo de evaluación de riesgos, y se definen las métricas necesarias. Básicamente, se mantienen los pasos incluyéndose uno intermedio, que se corresponde al modelo de evaluación. La primera vez que se realiza la identificación de riesgos, se construye el modelo, luego, éste se utiliza cada vez que se realiza seguimiento. A su vez, cuando se realizan nuevas identificaciones de riesgos, el modelo se actualiza contemplando los nuevos elementos detectados y las métricas necesarias.

8. Caso de Estudio

Como caso de estudio, describiremos parte del proceso de software del Proyecto de fin de carrera SMC de la Universidad ORT Uruguay, 2001-2002. Este proyecto, tiene como objetivos desarrollar un sistema de automatización del manejo de los documentos que tienen participación en el sistema de mejoramiento de la calidad de la empresa cliente.

Desde el comienzo, el cliente y el grupo de proyecto presentaron un conjunto de características que se tradujeron en factores de riesgo. Dados estos altos factores de riesgo, sumados a otros surgidos a partir de las restricciones del producto, se consideró utilizar un proceso basado en la gestión de los riesgos. Las toma de decisiones, se realiza tomando como entrada el análisis de los riesgos y la evolución de los mismos. De hecho, se toman en cuenta para planificar y decidir sobre aspectos que van desde las relaciones interpersonales y ciclo de vida, hasta los planes de versiones pre-release.

En vista de este objetivo, se decidió gestionar a lo largo de todo el proyecto, el conjunto de riesgos más importantes, realizándose además la gestión para los riesgos específicos que pueden surgir en cada fase. Para que esto sea posible, la gestión de riesgos debe ser un proceso lo más sencillo y automático posible, que se pueda repetir muchas veces, y preciso, para poder cuantificar y medir las variaciones. Para los factores de riesgo más importantes, se definió el conjunto de riesgos posibles, y se vincularon a un conjunto de métricas para automatizar el proceso de seguimiento. Las métricas nos brindan esa precisión que se estaba buscando.

A modo de ejemplo, vamos a describir la descomposición del factor de riesgo “Poca experiencia de los integrantes del grupo de proyecto”, cuyo esquema se puede ver en la Figura 6.

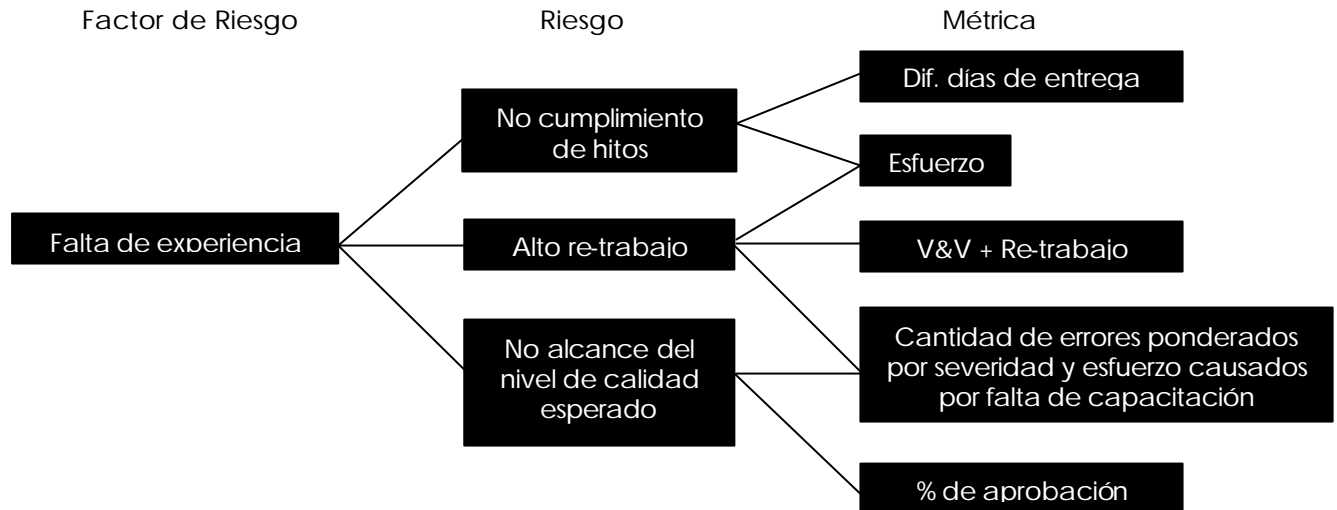


Figura 6: Ejemplo

Para la priorización de riesgos se optó por usar el modelo de Karolak, en especial definiéndose para el impacto del riesgo un rango de valores de 1 a 3, incrementándose de a 1. Para cada métrica se definió un mapeo entre los valores esperados y el impacto de la variable correspondiente.

Por ejemplo, para la métrica “Diferencia de días de entrega” con respecto al riesgo “No cumplimiento de hitos” se definió que:

Si dif ≤ 2	→ Impacto(Cronograma) = 1
Si dif > 2 y dif ≤ 5	→ Impacto(Cronograma) = 2
Si dif > 5	→ Impacto(Cronograma) = 3

Para la métrica “Cantidad de errores ponderados por severidad y esfuerzo causados por falta de capacitación” y el riesgo “Alto re-trabajo” se definió la siguiente relación, tomando en cuenta ciertas consideraciones con respecto a las ponderaciones de los problemas que van más allá del alcance de este ejemplo:

Si cant < 2	→ Impacto(Esfuerzo) = 1
Si cant ≥ 2 y cant < 4	→ Impacto(Esfuerzo) = 2
Si cant ≥ 4	→ Impacto(Esfuerzo) = 3

Al principio de un proyecto, no siempre se van a conocer todas las posibles métricas que pueden llegar a interesar, por lo cual es necesario, además de definir qué métricas se desean calcular, determinar qué datos se van a recolectar. En etapas tempranas del proyecto se decidió programar un conjunto de métricas básicas y recolectar una serie de datos en planillas que debían ser completadas por cada integrante del grupo. Cuando se realizó el plan de métricas se tuvieron en cuenta las instancias de evaluación de riesgos de modo que los datos estuvieran disponibles al momento de realizar estas actividades.

De las métricas mencionadas en el ejemplo, en una primera instancia fueron definidas la diferencia de días de entregas de productos, y la medición de esfuerzo. La métrica de “% de aprobación” fue incorporada más adelante por el líder de SQA, mientras que las métricas de “Verificación & Validación + Re-trabajo” y de “Cantidad de errores ponderados por severidad y esfuerzo” fueron definidas específicamente para ser incorporadas al marco de evaluación de riesgos.

Como se mencionó anteriormente, para que el proceso sea lo más automatizado posible, se crearon herramientas para cumplir este fin, buscando que se adapten dinámicamente a las métricas que se fueran incorporando, minimizando el esfuerzo extra y basándose siempre que sea posible en el conjunto de datos recolectados.

9. Conclusiones

A partir de la aplicación de algunos de los conceptos de los modelos teóricos de calidad de software y las metodologías de gestión de riesgos y de métricas, se plantea una metodología que combina lo anterior y un marco que sirve como herramienta auxiliar para la gestión de riesgos.

El modelo y metodologías propuestos son flexibles, dado que no están ligados a metodologías específicas, permitiendo a los gerentes utilizar las que más se adapten a las necesidades de sus proyectos. Además, son aplicables a lo largo del ciclo de vida del proyecto. A su vez, el uso de métricas otorga objetividad a los resultados obtenidos, lo cual es muy útil a la hora de realizar proyecciones sobre los riesgos, y tomar decisiones o plantear estrategias en consecuencia.

Referencias Bibliográficas

- [Boehm, 1989] - *Software Risk Management*. B. Boehm. IEEE Computer Society Press. 1989.
- [Boehm, 1995] - *Cost Models for Future Software Life Cycle Processes: COCOMO 2.0*. B. Boehm. IEEE Computer Society Press. 1995.
- [Fenton, 1997] - *Software Metrics. A Rigorous & Practical Approach*, Segunda Edición. Norman E. Fenton. Shari Lawrence Pfleeger. PWS Publishing Company. 1997.
- [Hyatt, 1996] - *A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality*. Lawrence E. Hyatt. Linda H. Rosenberg.
http://satc.gsfc.nasa.gov/support/STC_APR96/quality/stc_qual.PDF. 1996.
- [Karolak, 1996] - *Software Engineering Risk Management*. Dale W. Karolak. IEEE Computer Society Press. 1996.
- [Möller, 1993] - *Software Metrics. A Practitioner's Guide to Improved Product Development*. K. H. Möller. D. J. Paulish. Chapman & Hall. 1993.
- [Oman, 1997] - *Applying Software Metrics*. Paul W. Oman. Shari Lawrence Pfleeger. IEEE Computer Society Press. 1997.
- [Pfleeger, 1998] - *Software Engineering: Theory and Practice*. Shari Lawrence Pfleeger. Prentice Hall, Inc. 1998.
- [PMI, 1997] - *A Guide to the Project Management Body of Knowledge*. Standards COMMITTEE. PMI Publishing Division. 1997.
- [Pressman, 1998] - *Ingeniería de Software. Un enfoque práctico*, Cuarta Edición. Roger S. Pressman. McGraw Hill. 1998.
- [Schulmeyer, 1999] - *Handbook of Software Quality Assurance*, Tercera Edición. G. Gordon Schulmeyer. James I. McManus. Prentice Hall PTR. 1999.
- [SEI, 2002] - *Risk Management FAQ*. SOFTWARE ENGINEERING INSTITUTE.
<http://www.sei.cmu.edu/programs/sepm/risk/risk.faq.html>. 2002.